Proceedings of the 2nd International Conference on Trends in Electronics and Informatics (ICOEI 2018)
IEEE Conference Record: # 42666; IEEE Xplore ISBN:978-1-5386-3570-4

Automated Bug Reporting System In Web Applications

Yashika Sharma, Shatakshi, Palvika, Arvind Dagur, Rahul Chaturvedi
Department of Computer Science and Engineering
Krishna Engineering College, Ghaziabad, UP, India

***Abstract*— Bug reporting system is the approach for tracking and reporting bugs of a web application. It is the system which filters the duplicate bugs and generates the bug report. Bug reporting system permits individual or group of developers to observe and scan the bugs in their product adequately. The bug reporting system can greatly raise the work rate and responsibilities of particular working stiff by accommodating a recorded progress and optimistic response for pleasing performance. Manually generated the bug report is a time-consuming process and it is very expensive to maintain. In this paper, we designed a framework that automatically generates the bug report. It provides the facility of report logging, reusable method and multiple browser support which helps to reduce the human effort and time requires to performing regression testing.**

I. INTRODUCTION

As software projects turn out to be more expansive, complex and large. It comes to be more arduous to perform regression testing and generate the proper bug report or fix the bugs before shipping. Many organization uses bug tracking systems (e.g. Bugzilla and Zira) to manage bug resolution[11]. Good or typical bug reports consist of a useful information instead of a bug description and summary information.

A bug report contains all the details needed to document, report and remove problem arise in web applications. It is an approach that observes web application bugs and generates a bug report. Bug reporting system grant to report, store the document, manage, assign, close and acquire the bug reports. Those bug reporting systems used by utmost open source software projects, grant end-users to insert bug report directly. Other systems are used internally in the organizations for the purpose of software development.

A bug reporting system is generally an essential element of a satisfactory software development infrastructure and regular use of a bug or issue reporting system consider one of the "sign of an honourable software team". A large element of a bug reporting system is a database that tracks and records the information about bugs which is known [10]. The information may include the bug reported time, its safety, the defective program behaviour and includes the identity of the person who reported it.

Many organizations consider bug reports to direct corrective maintenance activity [2]. Manually bug report generating is a time taking process but once a bug report generates then within a short time interval, assigning it to a suitable developer. To enhance the quality of software by granting the users to comments [5]. Bug reporting system allows testers to report a track, describe, classify and comment on a bug report.

In today's scenario, most of the software applications are written as a web-based application run on the internet browser. In automation testing, the repeatable test cases are executed through a software testing tool. There is a number of open sources tools are freely available like selenium. We use selenium in this proposed model. There are many benefits to test the web application automatically. Automation testing requires tools, test script and software.

Good and efficient software projects are forced to ship with all known, unknown, pass, failed, skipped test cases. They lack the development and maintenance resources to deal with every defect. For e.g. "In 2005, one Mozilla developer claimed that every day, almost 300 bugs appear that need triaging. This is far too much for only the Mozilla programmers to handle" [3]. It is difficult to identify how many numbers of test cases are a pass, skip, fail and duplicate. Previous studies report that as many as 36% bug reports were duplicates or otherwise invalid [3].

We suggest a technique or a framework reduce human effort and time taken to perform regression testing and generate a bug report using selenium tool. The resultant report consists a total number of test case execute, a number of passes, fail, skip and time taken to execute these test case. The report also consists the name of the tester and the type of the operating system used. In our experiment, we apply a different type of test cases to perform regression testing on web applications through selenium and experimentally formalize its predictive power. We evaluate our approach's efficiency as a filter and its capability to report the bug.

The primary contributions of this paper are:

- It concentrates on automatically test the web application generates a bug report which contains information about fail, pass, and skip test cases.
- It filters the duplicate bugs.
- Mail functionality provides by this system model.

The format of this paper is as follows. Section 2 introduces related work. In Section 3, we validate our model, paying

deliberate attention to the proposed methodology of the project. In Section 4, we analyze the result and simulation of our model. Finally we conclude in Section 5 followed by references.

II. Related Work

In current bug reporting systems, the tester does not generate effective bug report which contains all required information needed by developers. Without this information fixing of bug is a difficult task. Life cycle cost of a software system[6] increase day by day.

Akhilesh Babu Kolheri, K-Tameezuddin, Kalpna Gudikardula define four fundamental directions, to enhance the effectiveness of bug tracking system. In this model the developer prototype application, by capturing essential information from user tracks the bug effective and resolve the bug quickly [8].

Sandeep Singh predicts a system to analyze different bug tracking tools and suggest a new set of selection criteria that gives a more satisfying result, as the industry needs to choose the better tool among the possible set of tools [9].

Anvik et al. propose a model by using support vector machine and text categorization which automatically allot bug reports to an appropriate developer. They claim that their proposed model can help a human triage by suggesting a set of developers for each incoming bug report [3].

WeiB et al. predict the fixing effort i.e. the effort required for performing defect fixing activities [7]. They leverage historical contain information and existing bug databases. They use existing bug report databases for gathering prerecorded information of development cost data. By the use of textual similarity, they find related defect reports. In JBoss project, they use experimental validation which involves 600 defect reports.

Kim and Whitehead claim that the time required to perform bug fixing activities is useful software quality measure [8]. They define a system which measures the time taken to resolve the bugs in two different software projects and perform the whole testing activities including duplicate [12] and non-duplicate bugs and total completion time of a real bug.

To resolve existing problem we proposed a system that filter the duplicates bug, generate automated bug report which contains the facility of mail. It reduces the time and human effort to perform regression testing. This system also supports multiple browsers and maintains some mandatory negative bug report.

III. Proposed Model

Our objective is to establish a model of automated bug reporting system which performs regression testing automatically and generate the bug report. It also finds the duplicate reported bugs with the help of filtration process and after generating the bug report it will send that resultant report to the concerned person or the team whose mail id is given to the system.

"An automated bug reporting system" is an automation framework. This framework will consist of report logging, a reusable method, multiple browser support etc. which will help in automating any web application. With the help of this framework, the regression testing becomes easy and effective.

It is the set of instructions like coding specification, test-data approach, object archive analysis etc. which we proceed during automation scripting produce useful outcomes like rising code re-usage, superior portability, decrease script maintenance cost etc. It has been growing up to nullify the problem established in the practicing manual system. The software is tested repeatedly during the different phase of software development lifecycle to ensure quality. Regression testing will perform to uncover new software bugs in the existing system after each modification or changes in software or application. At the time of delivery of software, it will check the hardware configuration and tested on the different supported operating system.

Since all the testing processes are manual that increases time, cost and also do not give testing reports as much as efficient due to human error [4]. But with the help of the automation it will save time, cost and gives an effective testing report and there is no chance for human error as it is performed by tools and or scripts. With the help of this automated bug reporting system you can start the testing when you are not there to monitor the system but when you will come you have your bug report

By filtering the duplicate bugs automatically we save the time of developers and they will focus on other testing activities. The duplicate bugs create problems for the developers because the bugs are more in numbers but most of them are duplicate which wastes the time and effort of the developers and makes debugging harder. But this duplicate bug finding feature helps developers to tackle that kind of problems.

Since the developer gets the bug report generated by the automated bug reporting system which contains the properly mentioned bug details. By mailing report feature also saves time and effort of the testers and there is no need to report all the bugs one by one.

Our model acts as a backbone of the testing framework. We find bugs from the web application by giving test cases as input and generated bug report according to given the test cases. We elicit certain information from the different bug report. When any new bug arrives, our system uses the extract information to predict the status of currently arriving bugs. For finding bug reports we make a framework. Testers write their test cases for their respective web applications and make changes in a framework according to their need and web application flow. The test cases executed and match different conditions given by the testers and generate the bug report. The test cases are written in the selenium. This report shows the number of passes, failed, skip test cases and also give details about each and every failed test case. This report also contains the graphical representation of the status of pass, fail and skip test cases. It will show the time at which the test cases will start executing and how much time it will take for completion of test cases execution, the owner of the system, configuration of the machine on which test cases execute and

have a very effective and impressive user interface. The framework is flexible so that testers change it accordingly.

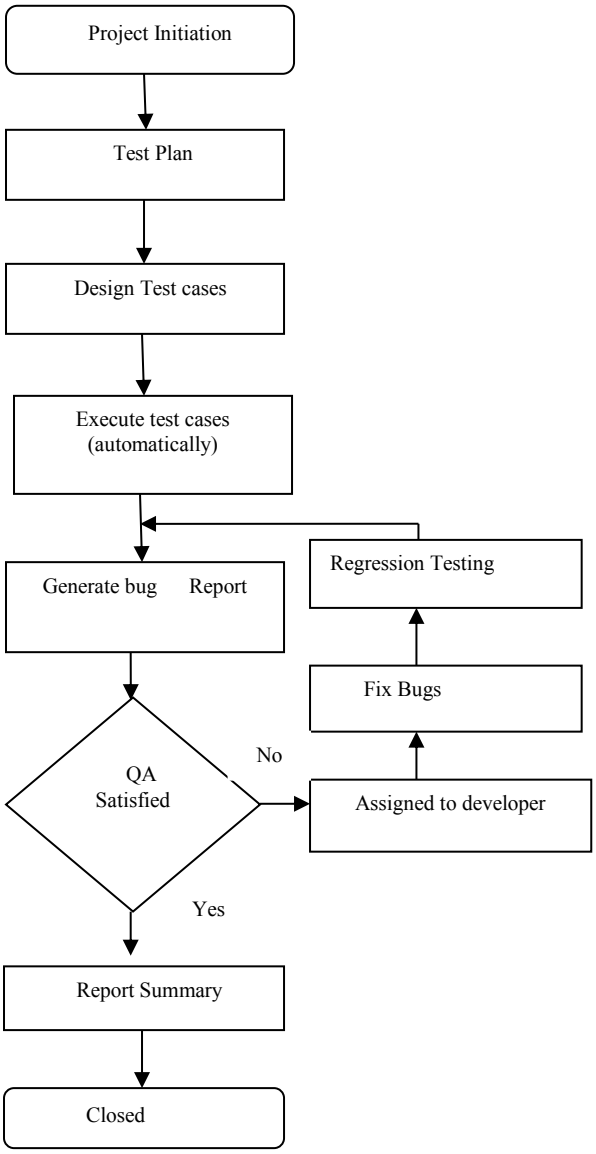

FIGURE-1 BUG REPORTING WORKFLOW

We use OOP's concept in our framework shown in Figure 1. We design a framework in which only the tester will give the test cases for the particular web application and the rest of the work will be done by the system starting from the testing of the web application to the bug report generation. Testers also use it as a layout or design to make our own framework for the testing. Today software needs quality because in our society there is a crisis of the software due to not develop qualitative software. With the help of this framework testers effectively test their web applications and achieve qualitative software.

Our developed system is not only for the newly developed web applications but also on legacy web applications. A key assumption of our model is that testers have knowledge about automatic testing tool i.e. selenium and there is one more assumption is that testers use this model to test one web application at a time. We cannot modify the generated reports and if we want to update then testers make changes in the test cases.

The automated bug reporting framework provides the faster processing of incoming bug because coefficient will calculate along time by using previously store bugs data. A newly generated bug report needs the features generation and extraction. The number of bugs reports are gathered to predicts the duplicate bug in future.

The software requirement specification is created at the completion of the different analysis chores[1]. The function and performance apportion system as a section of system engineering are processed by demonstrating information description and elaborate function and behavioral detailed description, other data opposite to requirement and conquer validation criteria.

FIGURE-2 PROTOTYPE OF RESULTANT BUG REPORT

Figure 2 shows a bug report prototype. Different colors are used to represent the resultant reports. The green color is used for representing pass test cases, red for failed test cases and blue for skipped test cases. This report will also contain the time and date when a tester will test the entire web application. It will also display the time duration for the execution of test cases of a particular web application.

This system will analyze the information about the duplicity of bugs. We also configure the mailing application with the framework which helps to send the reports just after its generation. This will tell each and every concerned person that

the test cases are executed and report (of the pass, fail, skip of test cases) is generated. And then time for the debugging of the bugs start by the developers.

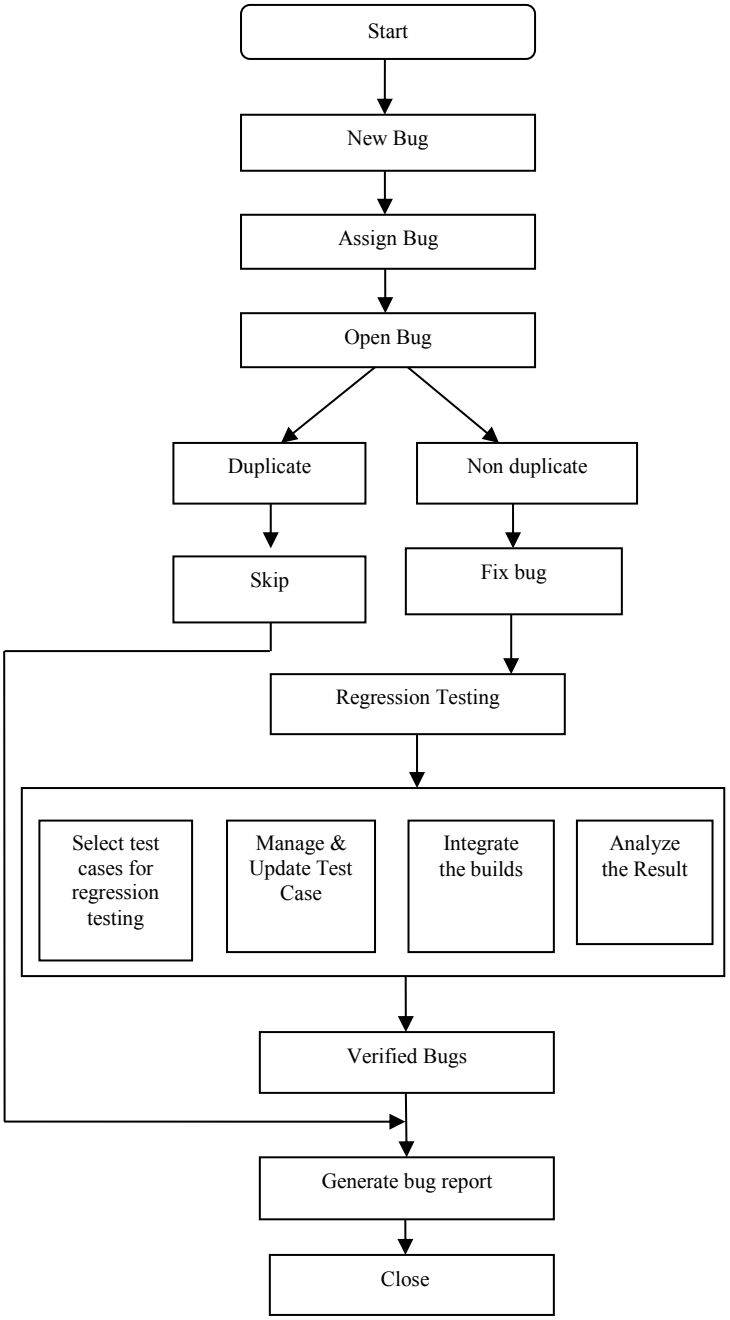

FIGURE-3 REGRESSION TESTING PERFORM ON THE BASIS OF CATEGORIZATION OF BUGS

Finally, we complete our model of Automated Bug Reporting System include the self-reported severity, number of associated patches and the relevant operating system and we consider the original bugs in the given dataset as shown in figure 3.

IV. RESULT AND SIMULATION

In the proposed model basically, we focus to reduce the time taken to perform regression testing on a web application. The analysis of this model shows that in the previous model, testers cannot effectively generate the bug report.

We analyze the performance of the model with the value of pass test case, failed test cases, skip test case and filtering of duplicate bugs as shown in figure 4.

FIGURE-4 GRAPHICAL REPRESENTATION OF BUG REPORT

This graphical representation of bug report helps to analyze the bugs without trouble as shown in figure 5. It is easy to understand because every case represent by different color. When the test cases apply to an application then the test cases split as pass, skip and failed. After the test the complete web application, this model automatically generates the bug report and provides the facility to mail it to the developers.

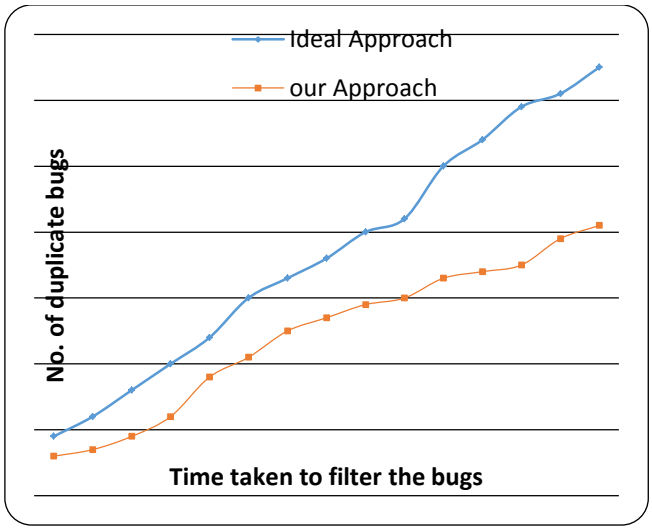

FIGURE-5 COMPARISON BETWEEN IDEAL APPROACH AND OUR APPROACH.

This graph in figure 5, shows the finding of duplicate bugs within a valid time period.

The previous existing model takes a large amount of time to perform regression testing manually but in automated testing, it takes a very short period of time.And in this model regression testing perform automatically as well as the summarized bug report will be generated automatically which show the information about the number of fails, pass and skip test cases.

In the existing model hard to analyze the bugs and time take to filter the bugs In this experiment, our approach is up to 8% better to test the web application and generate the bug report automatically than the previously published state.

V. CONCLUSION

We propose a model that automatically test the entire web application and perform the regression testing. This system is the best suited. This system is the best suited for an application which has multiple pages or states. It supports testing integrating cases (i.e, no need to create a new test class instance for every test method) and maintains each test cases in the different class file. It improves the reliability of web application and creates interactive documentation.

This proposed model will help to maintain the original code and code duplication which is crucial activity in test automation.

Thus, this model is capable to reduce the maintenance cost, development cost, human effort and time to perform regression testing by filtering the duplicate bugs.

REFERENCES

[1] P. Hooimeijer and W. Weimer. Modeling bug report quality. In Automated software engineering, pages 34-43, 2007.

[2] J. Anvik, L. Hiew, and G.C. Murphy.Coping with an open bug Repository. In OOPSLA workshop on Eclipse technology eXchange, pages 35-39, 2005.

[3] J. Anvik, L. Hiew, and G.C. Murphy. Who should fix this bug? In International Conference on Software Engineering (ICSEce), pages 361-370, 2006.

[4] C.V. Ramamoorthy and W.-T.T sai.Advances in software engineering. IEEE Computer, 29(10):47-58,1996.

[5] E.S.Raymond.The cathedral and the bazaar: musings on Linux and open source by an accidental revolutionary.Inf.Res.,6(4),2001.

[6] B.Boehm and V.Basili.Softer defect reduction. IEEE computer Innovative Technology for Computer Professions, 34(1):135-137, January 2001.

[7] C.WeiB, R.Premraj, T.Zimmermann, and A.Zeller. How long will it take to fix this bug? In Workshop on Mining Software Repositories, May 2007.

[8] S. Kim and J. E. James Whitehead. How long did it take to fix bugs? In International Workshop on Mining Software Repositories, pages 173-174, 2006.

[9] Sandeep Singh. Analysis of Bug Tracking Tools. International Journal of Scientific & Engineering Research, July 2013.

[10] Nicholas Jalbert and Westley Weimer. Automated Duplicate Detection for Bug Tracking Systems. In International Conference on Dependable Systems & Network, pages 52-61, 2008.

[11] Xin Xia, David Lo, Ying Ding, Jafar M. Al-Kofahi, Tien N. Nguyen, Xinyu Wang,. Improving Automated Bug Triaging with Specialized Topic Model. IEEE Transactions on Software Engineering.

[12] A. T. Nguyen, T. T. Nguyen, T. N. Nguyen, D. Lo, and C. Sun. Duplicate bug report detection with a combination of information retrieval and topic modeling. In Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering, pages 70–79. ACM, 2012.